

TP 04

**Exo 01 :**

Ecrire un programme en langage C permettant de remplir un tableau de **N** valeurs entières. Le nombre d'éléments **N** est au choix de l'utilisateur. Le programme doit ensuite afficher les éléments du tableau de la forme suivante :

Element Numero 0 = ...  
Element Numero 1 = ...  
Element Numero 2 = ...  
:  
Element Numero N-1 = ...

**Exo 02 :**

Ecrire un programme en langage C permettant de sauvegarder les notes d'un certain nombre d'étudiants **N**. Lorsque l'utilisateur introduit une valeur non valide (une note doit appartenir à l'intervalle  $[0,20]$ ), le programme doit informer l'utilisateur qu'il a fait une erreur de saisie. Le programme doit donner la chance à l'utilisateur de renouveler la saisie d'une note non valide. Une fois les notes des **N** étudiants sont sauvegardées, le programme doit afficher la meilleure note.

**Exo 03 :**

Écrire un programme en langage C qui remplit un tableau **T** de **N** entiers. Le programme doit copier ensuite toutes les composantes strictement positives dans un deuxième tableau **TPOS** et toutes les valeurs strictement négatives dans un troisième tableau **TNEG**. Les tableaux **TPOS** et **TNEG** doivent être affichés par le programme.

**Exo 04 :**

Ecrire un programme en langage C permettant de réaliser les traitements suivant :

1. Remplir deux tableau d'entiers **A** et **B**, triés dans l'ordre croissant. Lors de la saisie des valeurs de chacun des deux tableaux, à chaque fois, l'utilisateur doit saisir une valeur supérieure à la dernière valeur saisie.
2. Fusionner les deux tableaux **A** et **B** dans un troisième tableau **C** qui devrait être trié dans un ordre croissant.

Réécrire le même programme avec **A** trié dans l'ordre croissant et **B** est trié dans l'ordre décroissant.

**Exo 05 :**

Écrire un programme en langage C qui lit un entier **N** strictement positif. Le programme doit sauvegarder chacun des chiffres de l'entier **N** dans un tableau **T**. Le programme doit ensuite calculer et afficher le plus grand nombre qui peut être formé en utilisant les chiffres de l'entier **N**. Il suffit de trier le tableau **T** et d'utiliser ses éléments pour calculer le plus grand nombre.

**Exo 06 :**

Ecrire un programme en langage C permettant de remplir une matrice **MAT** (**N x M**) de type réel. Le programme doit ensuite calculer le minimum de chaque colonne et de le sauvegarder dans un tableau **MIN** de taille **M** de sorte que chaque élément du tableau **MIN** soit le minimum de la colonne lui correspondant. Le programme doit ensuite afficher la matrice **MAT** et le tableau **MIN**.

**Exemple :** MAT :

12	23	-6	56	98
23	12	-3	-1	1
3	87	29	9	5
11	40	33	36	25

MIN :

3	12	-6	-1	1
---	----	----	----	---

## Solutions

### Exo01 :

```
#include <stdio.h>
#include <stdlib.h>
const int taille = 50;
int main()
{ int T[taille];
  int n,i;
  do { puts("Donnez le nombre d'elements du
tableau");
    scanf("%d",&n);
  }while(n<1 || n>taille);
// la lecture du tableau T
for(i=0;i<n;i++){
  printf("T[%d] = ",i);
  scanf("%d",&T[i]);
}
// l'affichage des elements du tableau T
for(i=0;i<n;i++)
  printf("Element Numero %d = %d \n",i,T[i]);
return 0;
}
```

### Exo 02 :

```
#include <stdio.h>
#include <stdlib.h>
const int taille = 100;
int main()
{ float Note[taille];
  int n,i;
  do{ puts("Donnez le nombre d'etudiants");
    scanf("%d",&n);
  }while(n<1 || n>taille);
// la lecture du tableau T
for(i=0;i<n;i++){
  printf("Note[%d] = ",i);
  scanf("%f",&Note[i]);
}
if(Note[i]<0 || Note[i]>20){
  i--;
  printf("Erreur de saisie veuillez corriger \n");
}
// calcule de la meilleure Note
float m = Note[0];
for(i=1;i<n;i++)
```

```
if(Note[i]>m)
  m = Note[i];
printf("la meilleure Note est : %.2f \n",m);
return 0;
}
```

### Exo 03 :

```
#include <stdio.h>
#include <stdlib.h>
const int taille = 100;
int main()
{ int T[taille];
  int n,i;
  do{ puts("Donnez le nombre d'elements");
    scanf("%d",&n);
  }while(n<1 || n>taille);
// la lecture du tableau T
for(i=0;i<n;i++){
  printf("T[%d] = ",i);
  scanf("%d",&T[i]);
}
// creation des tableaux TPOS et TNEG
int TPOS[taille],TNEG[taille] ;
int NPOS=0,NNEG =0;
for(i=0;i<n;i++){
  if(T[i]>=0){
    TPOS[NPOS] = T[i];
    NPOS++;}
  else{
    TNEG[NNEG] = T[i];
    NNEG++;
  }
}
//affichage des tableaux.
printf("TPOS = {");
for(i=0;i<NPOS;i++)
  printf("%d, ",TPOS[i]);
printf("}\n");
printf("TNEG = {");
for(i=0;i<NNEG;i++)
  printf("%d, ",TNEG[i]);
printf("}\n");
return 0;
}
```

TP 04

**Exo 04 :**

```
#include <stdio.h>
#include <stdlib.h>
const int taille = 100;
int main()
{ int A[taille],B[taille];
int nA,nB,i;
do{ puts("Donnez le nombre d'elements de A");
scanf("%d",&nA);
}while(nA<1 || nA>taille);
// la lecture du tableau A
printf("A[0] = ");
scanf("%d",&A[0]);
for(i=1;i<nA;i++){
printf("A[%d] = ",i);
scanf("%d",&A[i]);
if(A[i]<A[i-1]){
printf("il faut saisir une valeur superieur a %d
\n",A[i-1]);
i--;
}
}

// la lecture du tableau B
do{ puts("Donnez le nombre d'elements de B");
scanf("%d",&nB);
}while(nB<1 || nB>taille);

printf("B[0] = ");
scanf("%d",&B[0]);
for(i=1;i<nB;i++){
printf("B[%d] = ",i);
scanf("%d",&B[i]);
if(B[i]<B[i-1]){
printf("il faut saisir une valeur superieur a %d
\n",B[i-1]);
i--;
}
}

// La fusion des tableaux A et B
int C[2*taille],n,iA=0,iB=0;
n = nA+nB;
for(i=0;i<n;i++){
if(A[iA]<B[iB]){
C[i] = A[iA];
```

```
iA++;
}else{
C[i] = B[iB];
iB++;
}
}
//affichage du tableau A
printf("A = {");
for(i=0;i<nA;i++)
printf("%d, ",A[i]);
printf("}\n");

//affichage du tableau B
printf("B = {");
for(i=0;i<nB;i++)
printf("%d, ",B[i]);
printf("}\n");

//affichage du tableau C
puts("Après la fusion");
printf("C = {");
for(i=0;i<n;i++)
printf("%d, ",C[i]);
printf("}\n");

return 0;
}
```

**Exo 05 :**

```
#include <stdio.h>
#include <stdlib.h>
const int taille = 100;
int main()
{ int n,c,t=0,i,j,chifre[20];
puts("Donner une valeur de n");
scanf("%d",&n);
while(n>0){
c = n%10;
n = n/10;
chifre[t] = c;
t++;
}
printf("chifre = {");
for(i=0;i<t;i++)
printf("%d, ",chifre[i]);
```

TP 04

```
printf("\n");
//le trie du tableau chiffre dans l'ordre decroissant.
for(i=0;i<t;i++){
    for(j=i+1;j<t;j++){
        if(chifre[j]>chifre[i]){
            c = chifre[i];
            chifre[i] = chifre[j];
            chifre[j] = c;
        }
    }
}
puts("le tableau chiffre apres le trie");
printf("chifre = {");
for(i=0;i<t;i++)
    printf("%d, ",chifre[i]);
printf("}\n");
// calcule du plus grand nombre n.
for(i=0;i<t;i++){
    n = n*10 + chifre[i];
}
printf("le plus grand nombre est: %d\n",n);

return 0;
}
```

**Exo 06 :**

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{ float MAT[20][20],MIN[20];
int n,m,i,j;
puts("Donner la taile de la matrice");
do{
    printf("n = ");scanf("%d",&n);
}while(n<1 || n>20);
do{
    printf("m = ");scanf("%d",&m);
}while(m<1 || m>20);

puts("donner les elements de la matrice");
for(i=0;i<n;i++)
    for(j=0;j<m;j++){
        printf("MAT[%d][%d] =
",i,j);scanf("%f",&MAT[i][j]);
    }
}
```

```
//affichage de la matrice
printf("MAT = ");
for(i=0;i<n;i++){
    for(j=0;j<m;j++){
        printf("%.2f ",MAT[i][j]);
    }
    printf("\n ");
}
// calcule du min de chaque colone
for(j=0;j<m;j++){
    MIN[j] = MAT[0][j];
    for(i=1;i<n;i++){
        if(MAT[i][j]<MIN[j])
            MIN[j] = MAT[i][j];
    }
}
//affichage du tablea MIN
printf("\nMIN = ");
for(i=0;i<m;i++)
    printf("%.2f ",MIN[i]);
return 0;
}
```